CLAIMS

What is claimed is:

1. A system for applying a persistence policy to override allocation of a resource based on application of a load balancing policy comprising:

first logic for determining if a persistence policy is applicable to a service request, and, if so, allocating a resource to the request based on application of the persistence policy; and

second logic for allocating a resource to the request based on application of a load balancing policy if the persistence policy is determined to be inapplicable as determined by the first logic.

2. The system of claim 1 wherein the first logic determines if a persistence policy is applicable to a service request having an originator through consideration of whether or not an allocation exists or recently expired for the originator the service request.

3. A system for allocating a resource to a resource request having an originator based on application of a persistence policy comprising:

first logic for determining whether an allocation exists or recently expired for the originator of the resource request, and, if so, identifying the resource which is the subject of the existing or recently expired allocation; and

second logic for allocating the resource, once identified, to the resource request.

4. The system of claim 3 wherein the resource request is derived from or represented by a packet.

5. A system for maintaining a data structure useful for allocating a resource to a resource request based on application of a persistence policy comprising:

first logic for making an entry in the data structure representing an allocation, upon or after implementation of the allocation, and time-stamping the entry with a time-stamp indicating the time when or about when the allocation is terminated; and

second logic for scanning the data structure, and deleting entries for which a time-out condition is determined to exist.

6. The system of claim 5 wherein the second logic has access to a current time, and determines that a time-out condition exists if the time-stamp value equals or exceeds the current time by a predetermined amount.

7. The system of claim 5 wherein the second logic repetitively scans the data structure.

8. The system of claim 5 wherein the second logic periodically scans the data structure.

9. The system of claim 6 wherein the predetermined amount is programmable.

10. A system for making an entry in a data structure representing an allocation, the data structure being useful for allocating a resource to a resource request based on application of a persistence policy, the system comprising:

first logic for deriving a first index from information relating to the resource request if such information is available, and using the first index to make an entry in the data structure representing the allocation if the first index is available; and

second logic for deriving a second index from information relating to the resource request, and using the second index to make an entry in the data structure representing the allocation.

11. The system of claim 10 wherein the data structure is a history table.

12. The system of claim 10 wherein the first logic derives each of the first and second indices by applying a hashing function to information derived from the resource request.

13. The system of claim 10 wherein the first logic derives the first index by applying a hashing function to a hashing key derived from a session or cookie identifier derived from a packet spawning the resource request.

14. The system of claim 10 where the second logic derives the second index by applying a hashing function to a hashing key derived from a client IP address derived from a packet spawning the resource request.

15. A system for making an entry in a data structure representing an allocation, the data structure being useful for allocating a resource to a resource request based on application of a persistence policy, the system comprising:

first means for deriving a first index from information relating to the resource request if such information is available, and using the first index to make an entry in the data structure representing the allocation if the first index is available; and

second means for deriving a second index from information relating to the resource request, and using the second index to make an entry in the data structure representing the allocation.

16. A system for accessing a data structure in order to allocate a resource to a resource request based on application of a persistence policy, entries in the data structure corresponding to allocated resources, the system comprising:

first logic for deriving a first index from information relating to a resource request if such information is available, using the first index to access the data structure and determine if an entry corresponding to the first index is available, and, if such an entry is available, allocating the resource corresponding to the entry to the resource request; and

second logic for deriving, if the first index or an entry corresponding to the first index is unavailable, a second index from information relating to the resource request, and using the second index to access the data structure and determine if an entry corresponding to the second index is available, and, if such an entry is available, allocating the resource corresponding to the entry to the resource request.

17. The system of claim 16 further comprising third logic for allocating, if an entry corresponding to the second index is unavailable, a resource to the request based on application of a load balancing policy or other persistence policy.

18. The system of claim 17 further comprising fourth means for using the first index to make an entry in the data structure corresponding to the allocation of claim 17 if such first index is available.

19. The system of claim18 further comprising fifth means for using the second index to make an entry in the data structure corresponding to the allocation of claim 17.

20. A method of applying a persistence policy to override allocation of a resource based on application of a load balancing policy comprising:

determining if a persistence policy is applicable to a service request, and, if so, allocating a resource to the request based on application of the persistence policy; and

allocating a resource to the request based on application of a load balancing policy if the persistence policy is determined to be inapplicable in the foregoing determining step.

21. A method of allocating a resource to a resource request based on application of a persistence policy, the request having an originator, comprising:

determining whether an allocation exists or recently expired for the originator of the resource request, and, if so, identifying the resource which is the subject of the existing or recently expired allocation; and

allocating the resource, once identified, to the resource request.

22. The method of claim 21 wherein the resource request is spawned by a packet.

23. A method of maintaining a data structure useful for allocating a resource to a resource request based on application of a persistence policy comprising:

making an entry in the data structure representing an allocation, and time-stamping the entry with a time-stamp indicating the time when or about when the allocation is terminated; and

scanning the data structure, and deleting entries for which a time-out condition is determined to exist.

24. The method of claim 23 further comprising determining that a time-out condition exists if the time-stamp value equals or exceeds a current time by a predetermined amount.

25. The method of claim 23 further comprising repetitively scanning the data structure.

26. The method of claim 23 further comprising periodically scanning the data structure.

27. The method of claim 24 wherein the predetermined amount is programmable.

28. A method of making an entry in a data structure representing an allocation, the data structure being useful for allocating a resource to a resource request based on application of a persistence policy, the method comprising:

deriving a first index from information relating to the resource request if such information is available;

using the first index to make an entry in the data structure representing the allocation if the first index is available;

deriving a second index from information relating to the resource request; and

using the second index to make an entry in the data structure representing the allocation.

29. The method of claim 28 wherein the data structure is a history table.

30. The method of claim 28 further comprising deriving each of the first and second indices by applying a hashing function to information derived from a packet spawning the resource request.

31. The method of claim 28 further comprising deriving the first index by applying a hashing function to a hashing key derived from a session or cookie identifier derived in turn from a packet spawning the resource request.

32. The method of claim 28 further comprising deriving the second index by applying a hashing function to a hashing key derived from a client IP address derived in turn from a packet spawning the resource request.

33. A method of making an entry in a data structure representing an allocation, the data structure being useful for allocating a resource to a resource request based on application of a persistence policy, the method comprising:

a step for deriving a first index from information relating to the resource request if such information is available;

a step for using the first index to make an entry in the data structure representing the allocation if the first index is available;

a step for deriving a second index from information relating to the resource request; and

a step for using the second index to make an entry in the data structure representing the allocation.

34. A method of accessing a data structure in order to allocate a resource to a resource request based on application of a persistence policy, entries in the data structure corresponding to allocated resources, the method comprising:

deriving a first index from information relating to a resource request if such information is available;

using the first index to access the data structure and determine if an entry corresponding to the first index is available;

if such an entry is available, allocating the resource corresponding to the entry to the resource request;

deriving, if the first index or an entry corresponding to the first index is unavailable, a second index from information relating to the resource request;

using the second index to access the data structure and determine if an entry corresponding to the second index is available; and

if such an entry is available, allocating the resource corresponding to the entry to the resource request.

35. The method of claim 34 further comprising allocating, if an entry corresponding to the second index is unavailable, a resource to the request based on application of a load balancing policy or other persistence policy.

36. The method of claim 35 further comprising using the first index to make an entry in the data structure corresponding to the allocation of claim 35 if such first index is available.

37. The method of claim 36 further comprising using the second index to make an entry in the data structure corresponding to the allocation of claim 35.

38. A method of accessing a data structure in order to allocate a resource to a resource request based on application of a persistence policy, entries in the data structure corresponding to allocated resources, the method comprising:

a step for deriving a first index from information relating to a resource request if such information is available;

a step for using the first index to access the data structure and determine if an entry corresponding to the first index is available;

a step for allocating, if such an entry is available, the resource corresponding to the entry to the resource request;

a step for deriving, if the first index or an entry corresponding to the first index is unavailable, a second index from information relating to the resource request;

a step for using the second index to access the data structure and determine if an entry corresponding to the second index is available; and

a step for allocating, if such an entry is available, the resource corresponding to the entry to the resource request.

39. The system of any of claims 1, 5, 10, 15 or 16, wherein the resource is a server.

40. The system of any of claims 1, 5, 10, 15 or 16, wherein the request is spawned by, represented by, or in the form of a packet.

41. The system of any of claims 1, 5, 10, 15 or 16, wherein an allocation results in, or corresponds to, a connection.

42. The system of any of claims 1, 5, 10, 15 or 16 implemented as one or more engines.

43. The method of any of claims 20, 21, 23, 28, 33, 34 or 38, wherein the resource is a server.

44. The method of any of claims 20, 21, 23, 28, 33, 34 or 38, wherein the request is spawned by, represented by, or in the form of a packet.

45. The method of any of claims 20, 21, 23, 28, 33, 34 or 38, wherein an allocation results in, or corresponds to, a connection.